

NOTE: THIS Dwg. S/B
ENHANCED, AS PER,
EG. 6, 055, 295

RADIATION GENERATION
SYSTEM

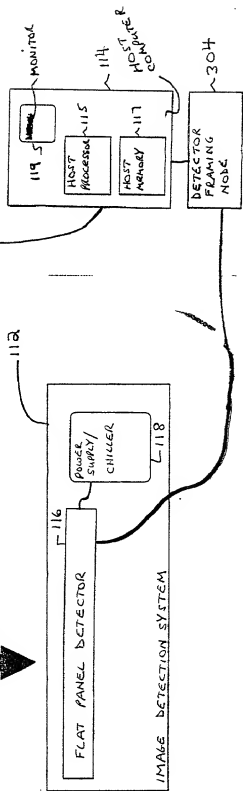
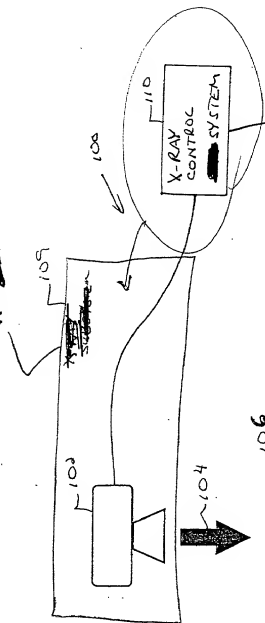


FIG. 1

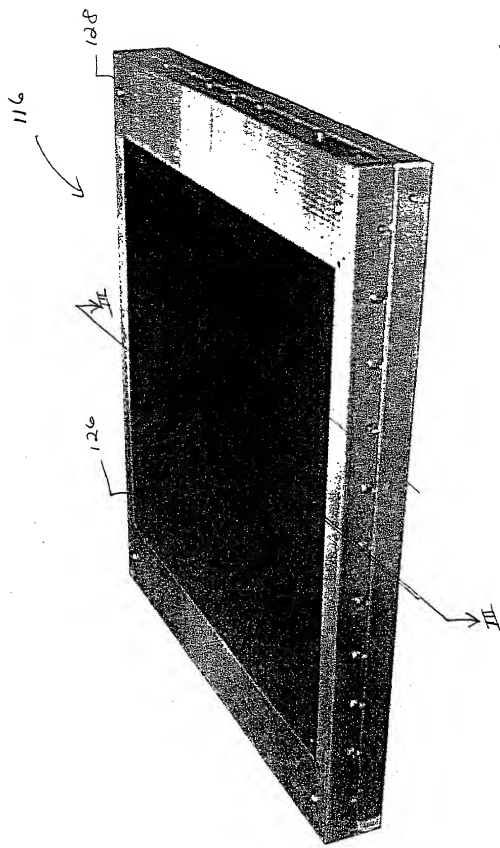


FIG. 2 (PRIOR ART)

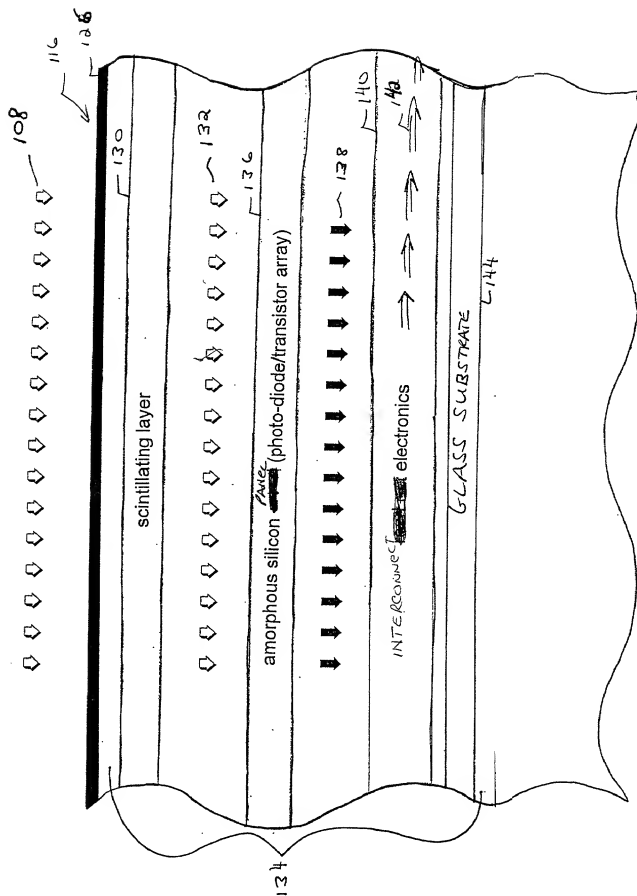


FIG. 3

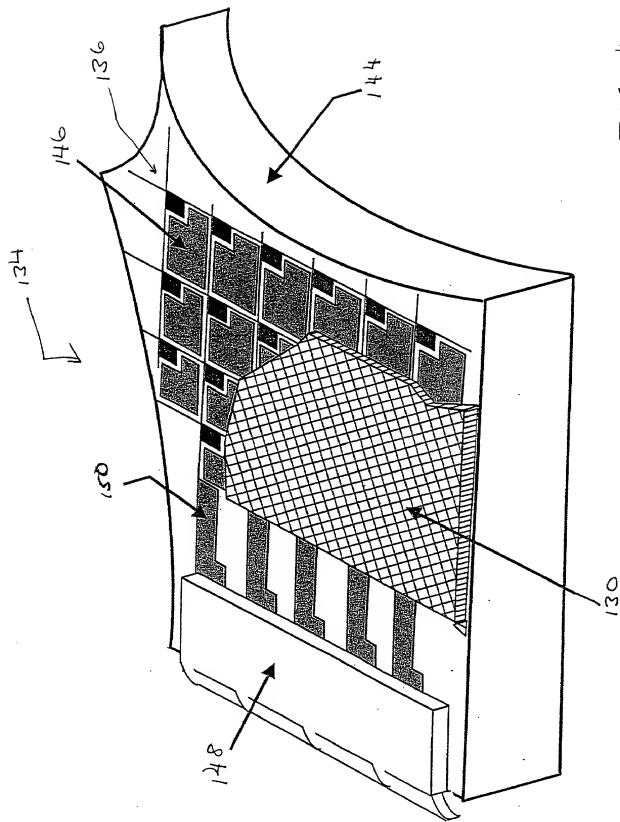


FIG. 4
(PRIOR ART)

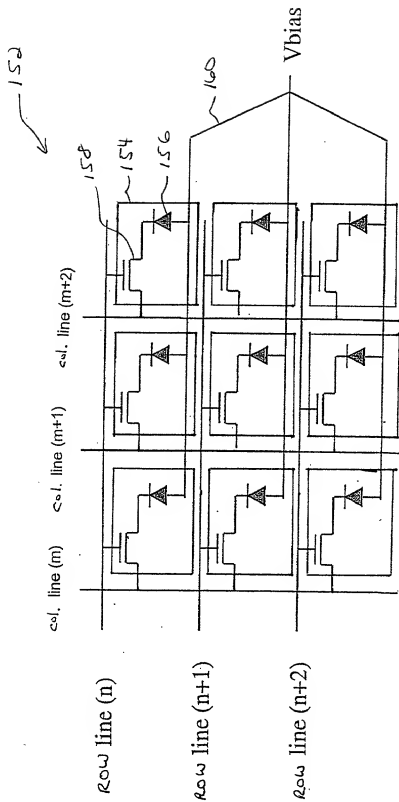


FIG. 5
(Prior Art)

FLAT PANEL DETECTOR

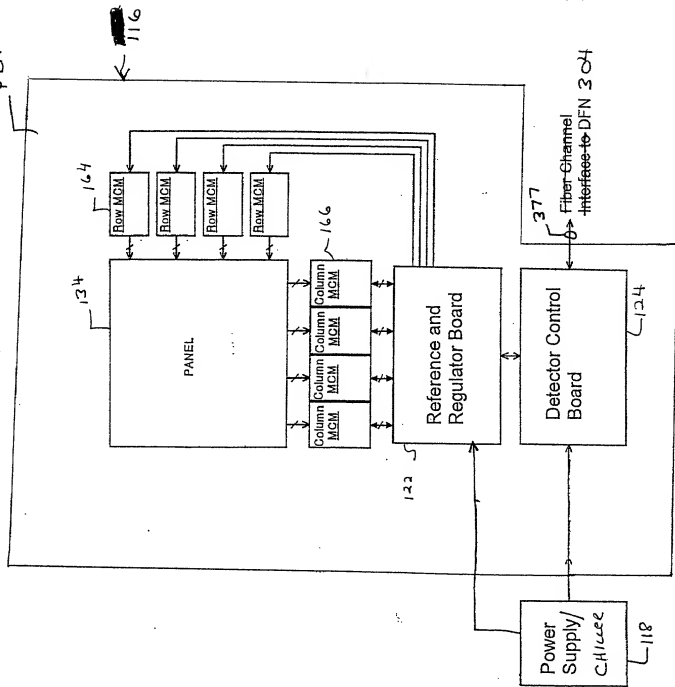


FIG. 6
(Prior Art)

FLAT PANEL DETECTOR

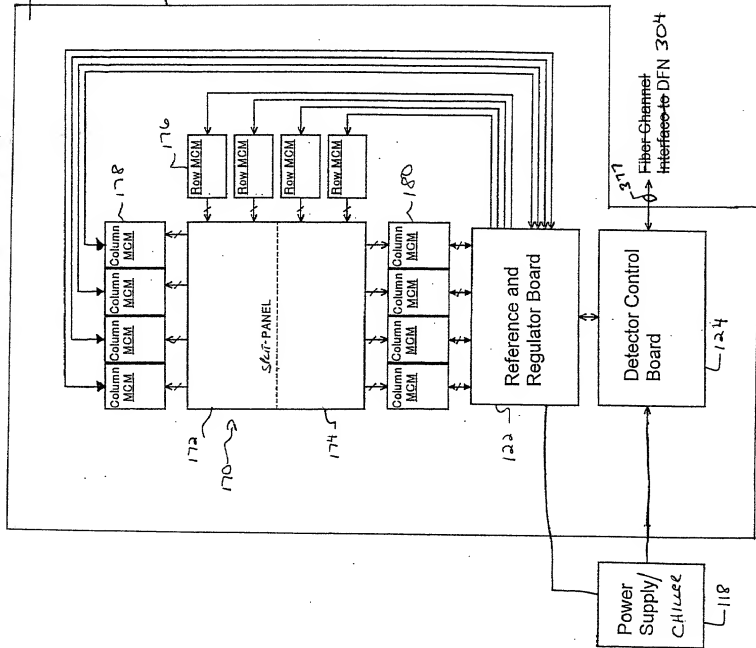
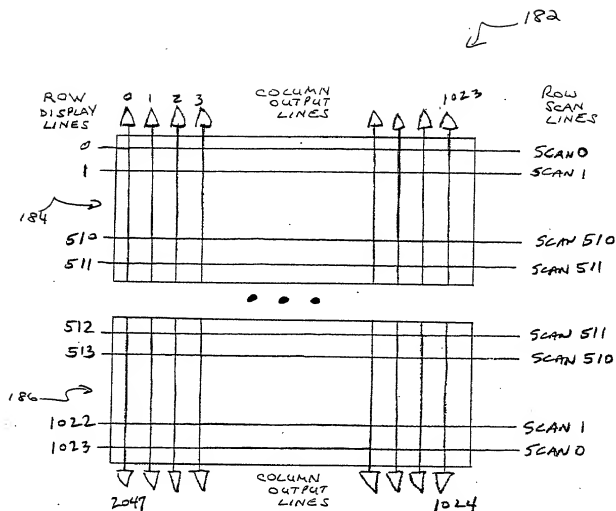


FIG. 7
(PRIOR ART)



CARDIAC/SURGICAL DIGITAL X-RAY PANEL

FIG. 8
(PRIOR ART)

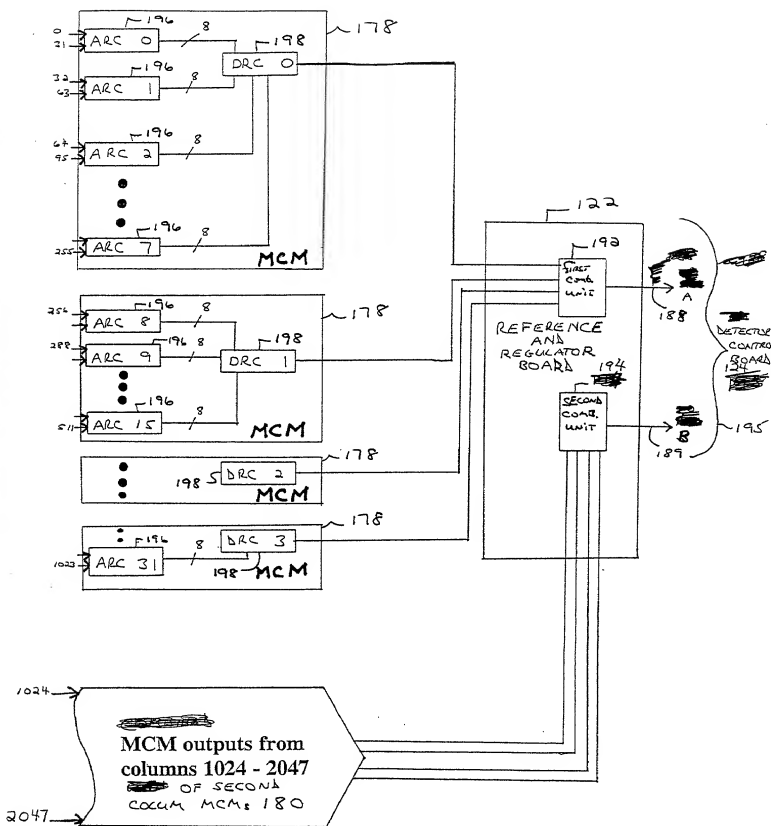


FIG. 9
(PRIOR ART)

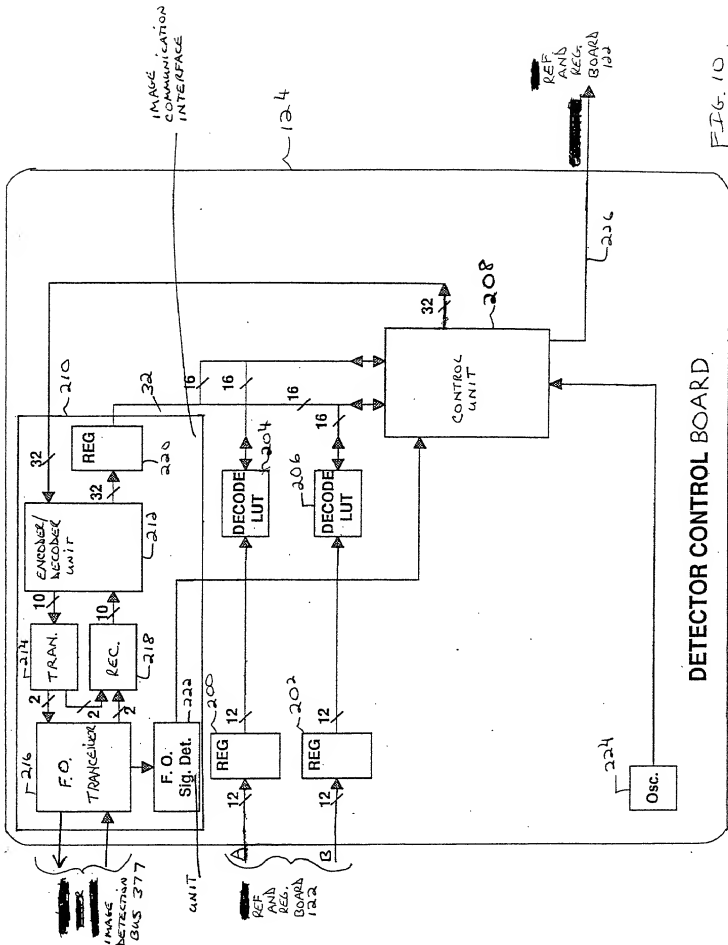
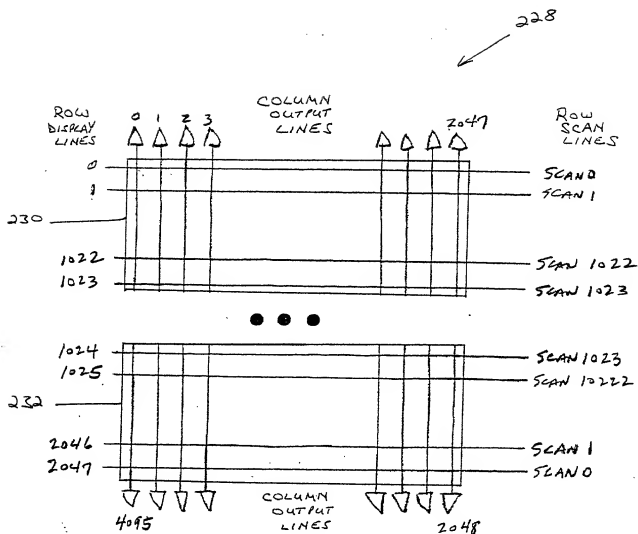


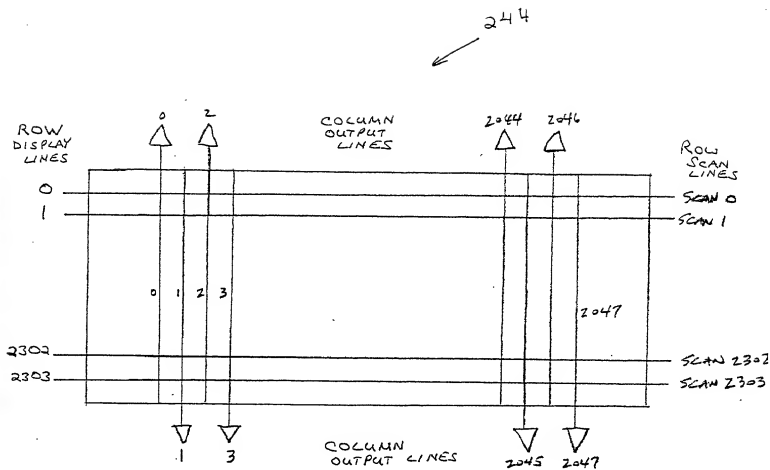
FIG. 10
(PRIOR ART)



RADIOGRAPHY DIGITAL X-RAY PANEL

FIG. 11
(PRIOR ART)

FIG. 13



MAMOGRAPHY DIGITAL X-RAY PANEL

FIG. 13
(PRIOR ART)

FLAT PANEL
DETECTOR

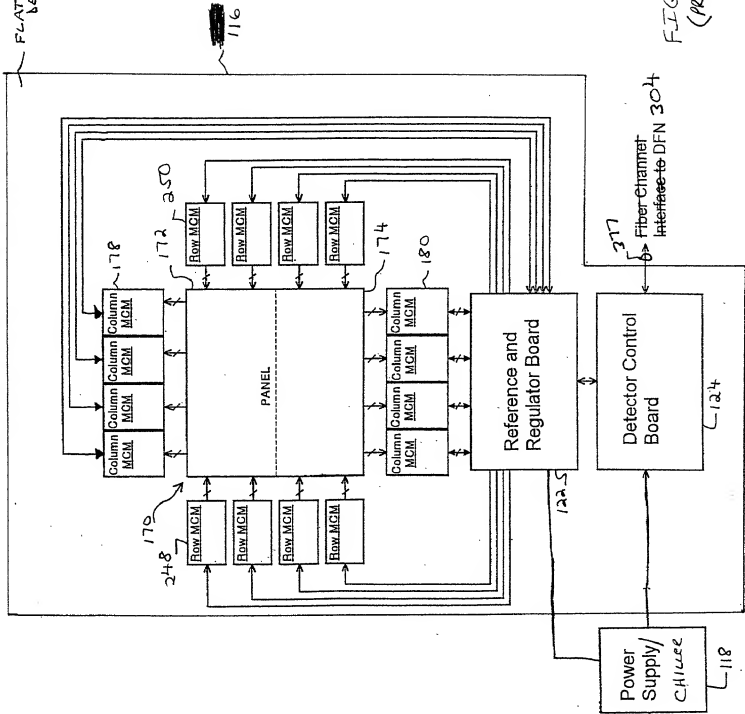


FIG. 14
(Prior Art)

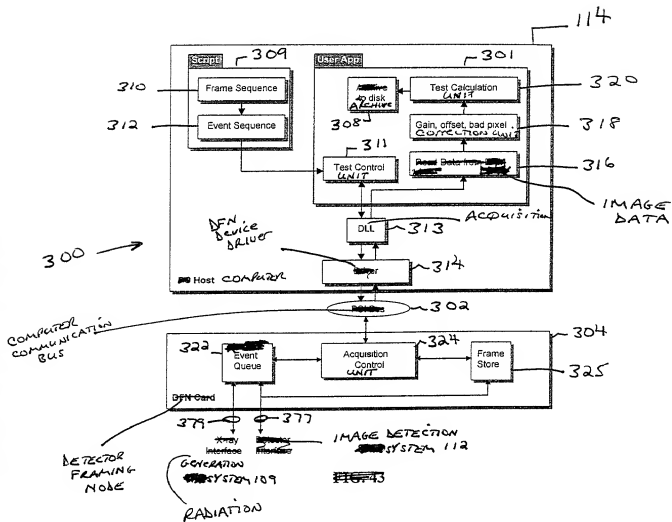


FIG. 15

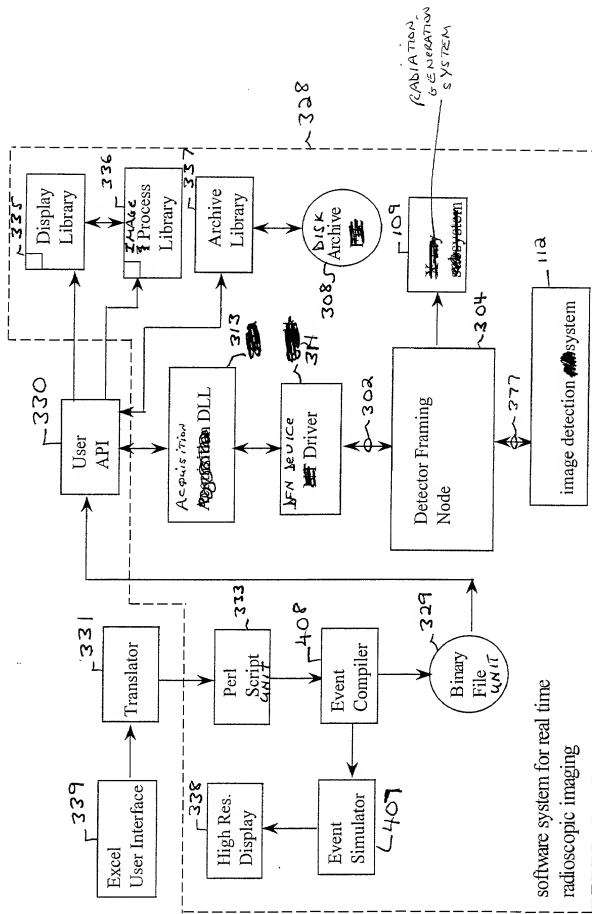


FIG. 16

IMAGE DETECTION
SYSTEM
112

~~IMAGE~~ GENERATION
SYSTEM
109

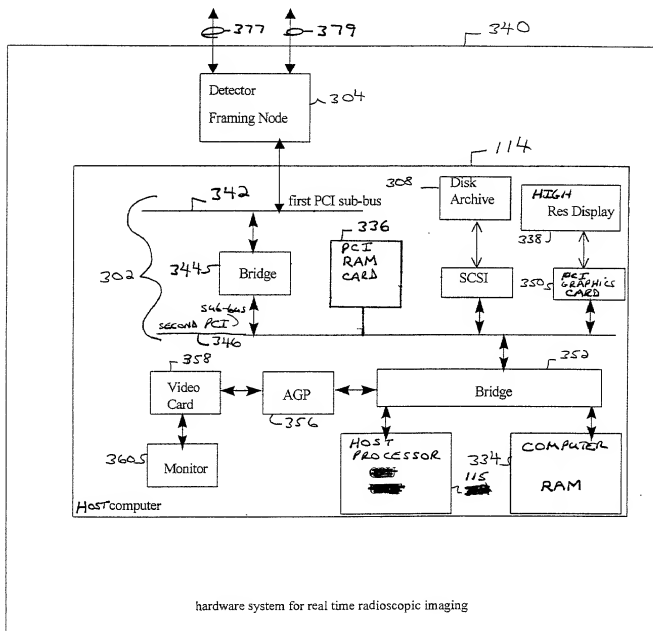


FIG. 17

DATE: 01-25-2001

FRAME BUFFER MEMORY

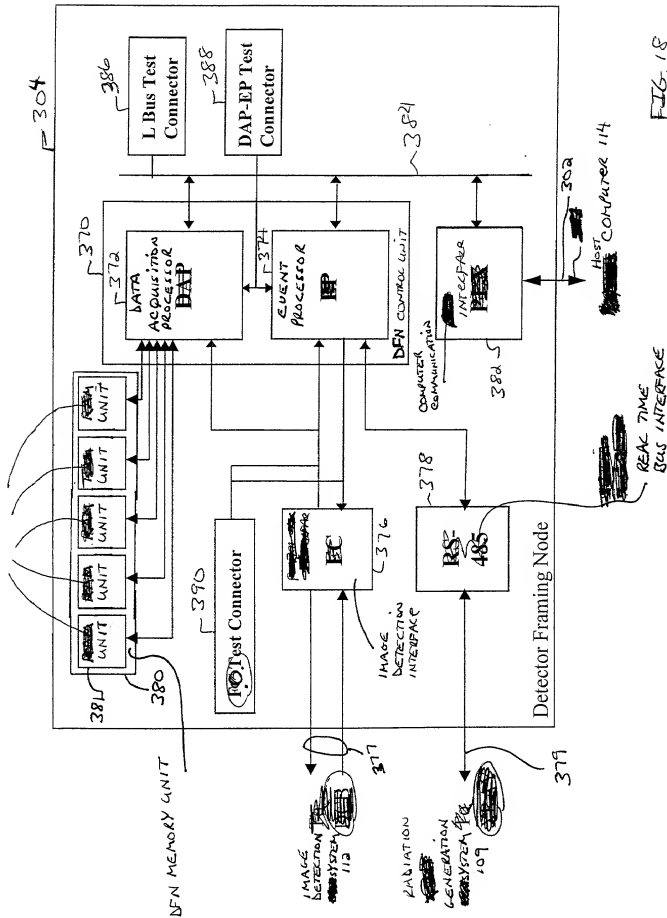


FIG. 18

Panel Setup	Real Time	(fm/sec)	length	Latency	memory	offset	gbr
Single Frame	Post Process	30	unlimited	< 5 frames	host	none	
Single Frame	Post Process	-	-	Delay ~ 1 sec	"	y	y
		-	-	Delay ~ 2 sec	"	y	y
Real Time	Real Time	R	Unlimited	< 5 frames	host	none	
Real Time	Real Time	R - X	Unlimited	< 5 frames	"	y	y
Real Time	Real Time	R - Y	Unlimited	< 5 frames	"	y	y

FIG-19

Modality	Image size	Frames Stored
Cardiac	1024 x 1024	host memory
Rad	2048 x 2048	200
Mammo	2304 x 2048	50
		44

FIG. 20

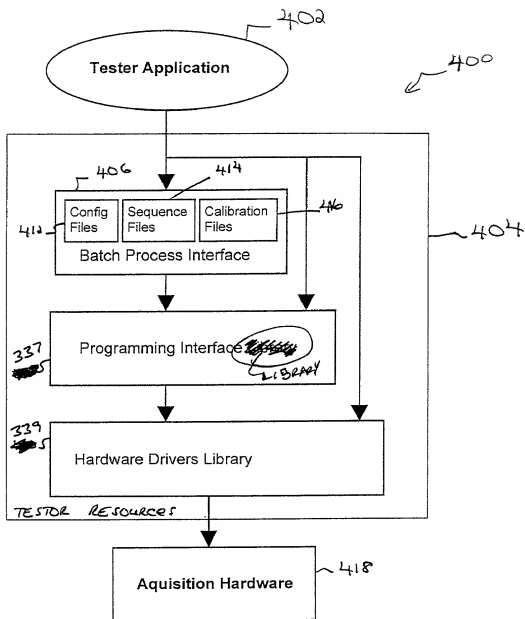


FIG. 21

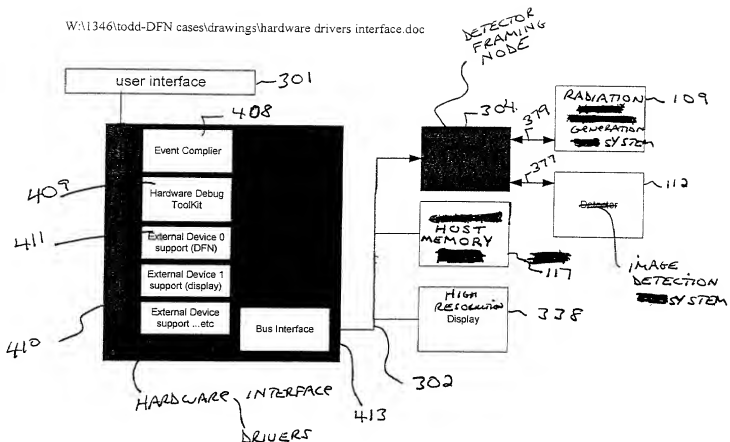


FIG. 22

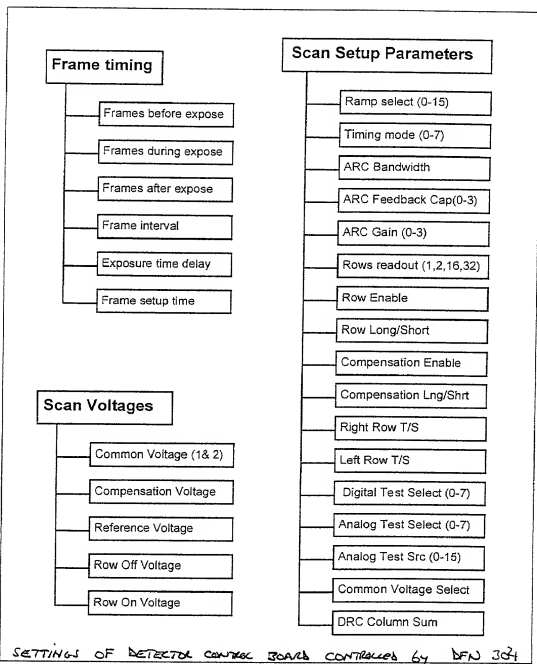


FIG. 23

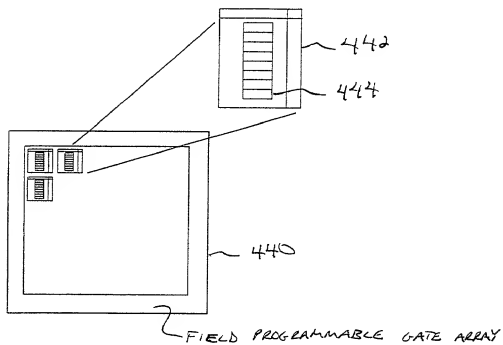


FIG. ~~23~~ 24

~~DETECTOR~~
CONTROL ROADS 124

EVENT
PROCESSOR

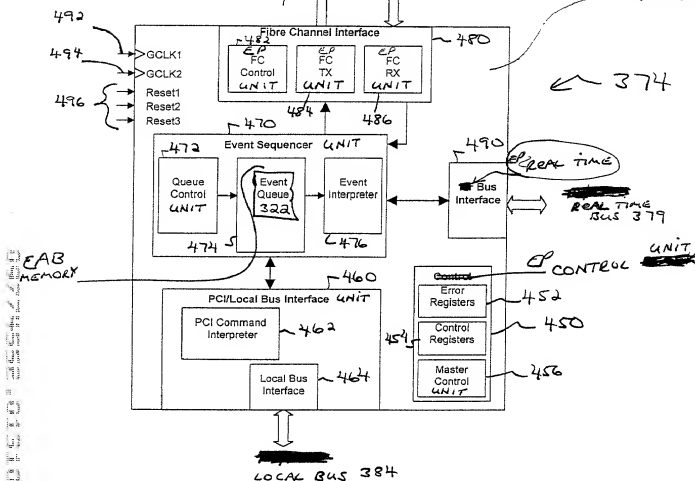


FIG. 25

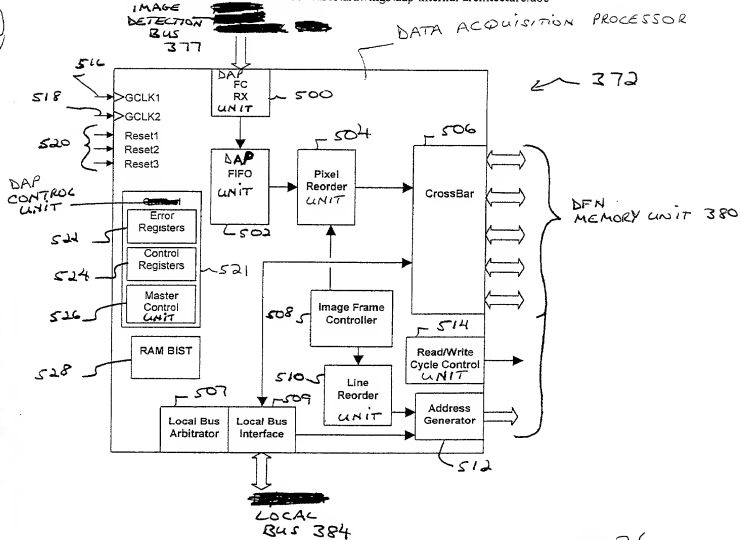
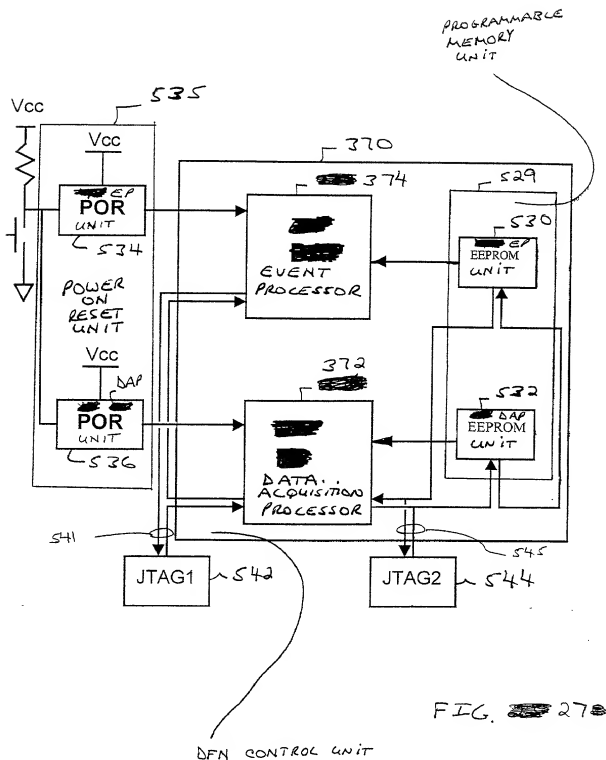
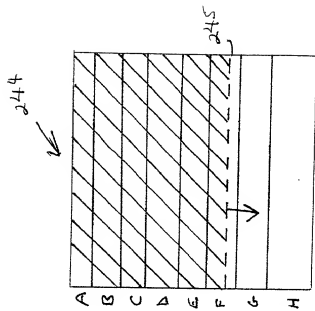
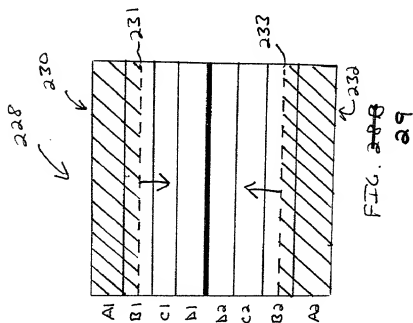
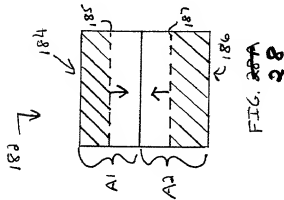


FIG. 26





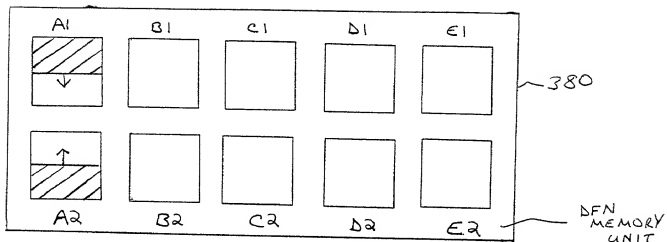


FIG. ~~30A~~ 31

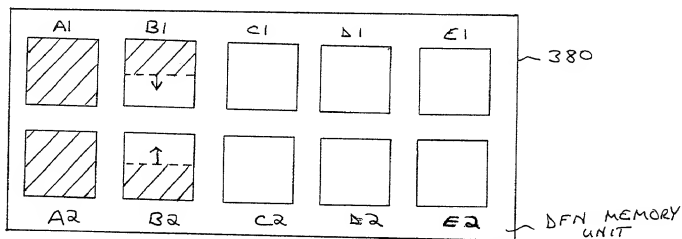


FIG. ~~30B~~ 32

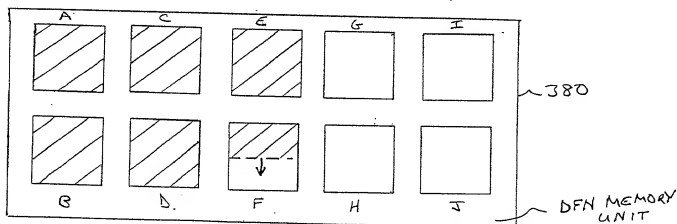


FIG. ~~30C~~ 33

334



A1	
A2	

FIG. 34

334



A1	
B1	
C1	
A1	
A2	
C2	
B2	
A2	

FIG. 35

334



A	
B	
C	
A	
E	
F	
G	
H	

FIG. 36

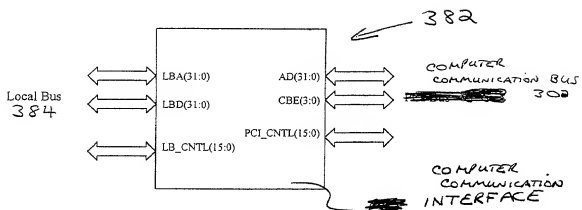


FIG. 37

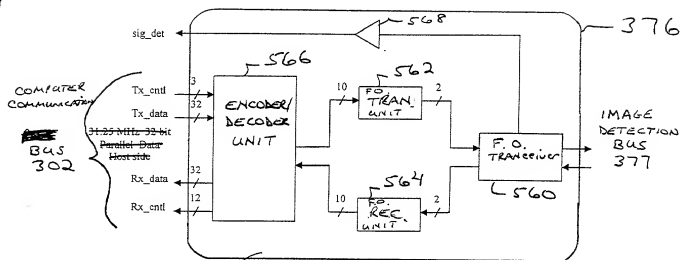


FIG. 38

IMAGE DETECTION INTERFACE

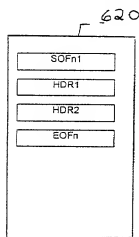


FIG. 39

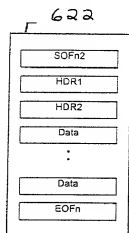


FIG. 40

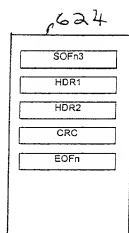


FIG. 41

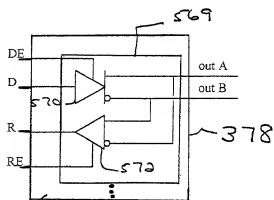
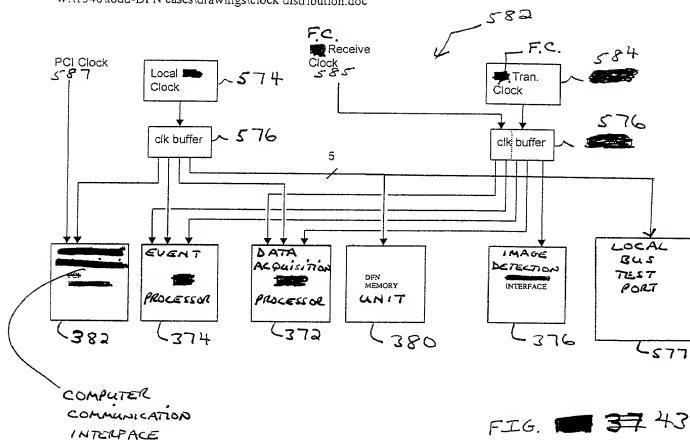


FIG. 42

REAL TIME
BUS INTERFACE



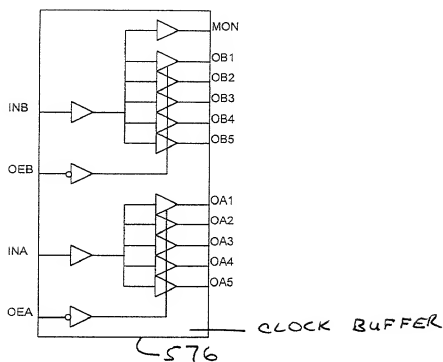


FIG. ~~44~~ 44

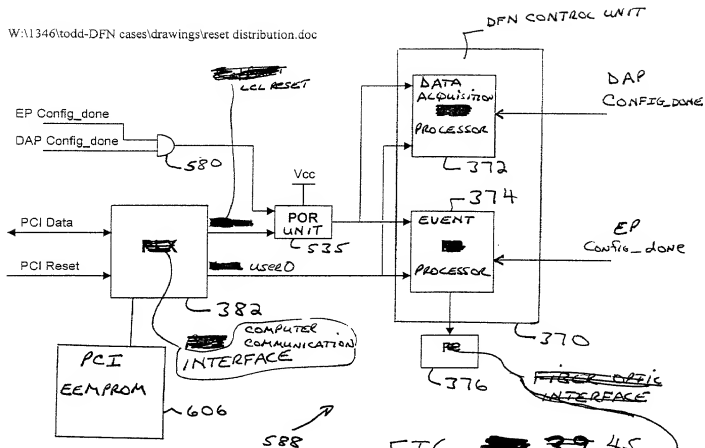
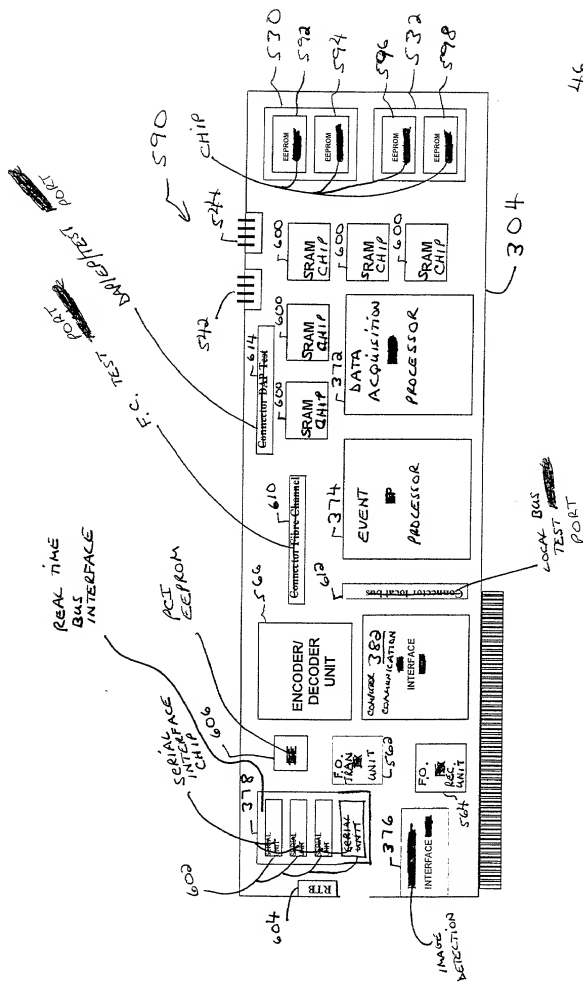


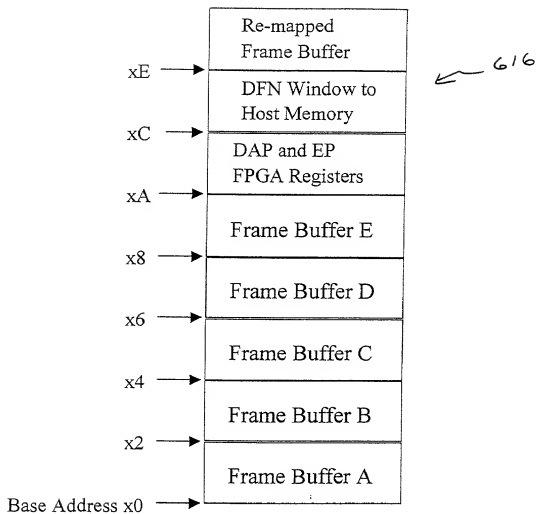
FIG. ~~39~~ 45

IMAGE DETECTION INTERFACE

~~COMPUTER COMMUNICATION INTERFACE~~



46
#46
FIG.



Mapping of 16 MByte PCI Address Space

FIG. 47

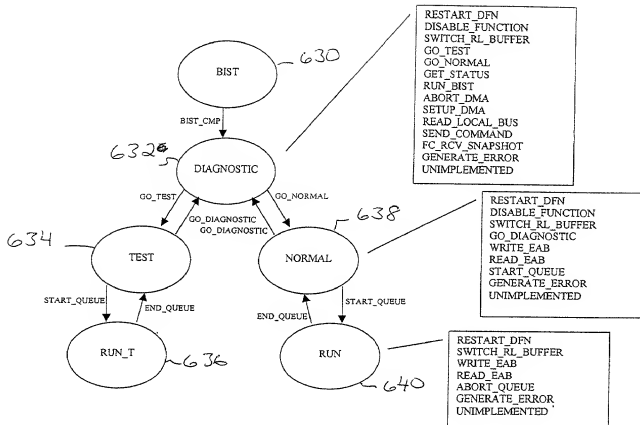


FIG. 48

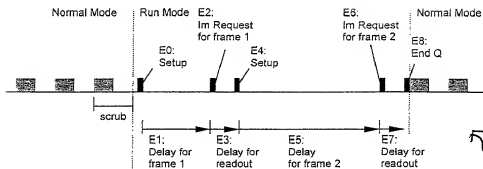


FIG. 49

Event Mnemonic	Event (showing size of arguments)	Op Code (hex)	Data (bytes)	Total (bytes)
Endq	Endq	14	0	1
Delay (1)	Delay (0xH ff ff ff)	10	4	5
Send (command, value)	Send (0xH ff ff ff, 0xH ff ff)	04	8	9
LoopKN (K, N)	LoopKN (0xH ff, 0xH)	0C	3	4
LoopKF (K, F)	LoopKF (0xH ff, 0xH ff ff)	0D	5	6
Wait (F)	Wait (0xH ff ff)	09	3	4
Flag (F)	Flag (0xH ff ff)	08	3	4

FIG. 50

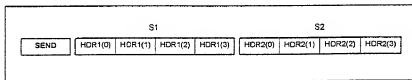


FIG. 51

Error Mnemonic	Description of Error
FC_TIMEOUT	Timeout expired with no ACK detected
FC_BAD_ACK	ACK did not match transmitted command
FC_EXTRA_ACK	Unexpected ACK received
FC_EXTRA_CMD	New Send event while waiting for ACK from previous Send
SIG_DETN	No input signal power on Fibre Channel (cable disconnected?)
RXERROR	Fibre Channel receiver detected bad data (defective chipset?)
WRDSYNCN	Fibre Channel Data link unsynchronized
CRXS(1)	Bad Received CRC detected (Fiber-optic cable problem?)
CRXS(3) and CRXS(2)	Bad order in link state machine (defective chipset?)

FIG. 52

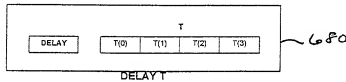


FIG. 53

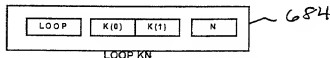
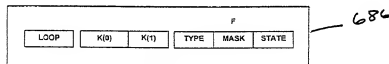


FIG. 54



LOOP KF

FIG. 55

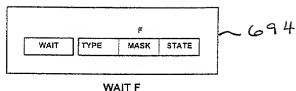


FIG. 56

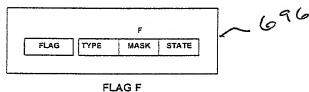


FIG. 57

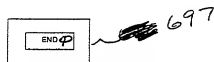


FIG. 58

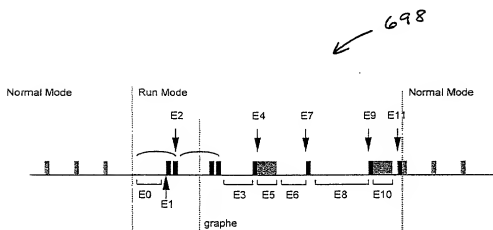


FIG. 59

E11	EndQ
E10	Delay 125 ms
E9	Send Im Request
E8	Delay 500 ms
E7	Flag RT2
E6	Delay 50 ms
E5	Delay 125 ms
E4	Send Im Request
E3	Delay 300 ms
E2	Loop 2, RT1
E1	Send Scrub
E0	Delay 300 ms

Event Queue

FIG. ~~60~~ 60

702

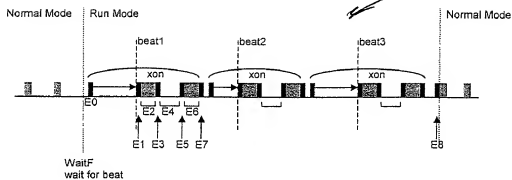


FIG. ~~61~~ 61

	EndQ
E7	LoopKF HF1
E6	Delay 23 ms
E5	Send Im Request
E4	Delay 10 ms
E3	Flag RT2
E2	Delay 23 ms
E1	Send Scrub Request
E0	Wait RT1

Event Queue

FIG. 62

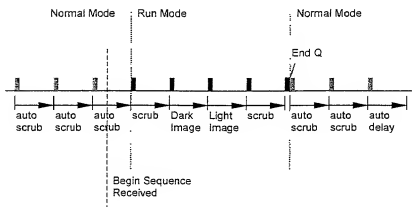


FIG. 63

```
sequence_begin ();

# define qv defaults:
%qvl = ('delay_qv' => 5000);

# call frame with qv's
frame_type1 (NULL, \%qvl, 1);

sequence_end ();
```

FIG. 64

```
sub frame
{
  $QVf = 'frame';

  %qv = ('delay_qv' => [10000]);
  %qp = ();

  compile_init(@_, \%qp, \%qv, $QVf);

  Delay('delay_qv');

  compile_finit();
}
```

FIG. 65

```
pDFN->DFNChangeQueueVariable
{
  (char *)SymName,    // variable name
  (char *)sndBuf,     // new value
  BufSize,           // num bytes to write
  (ULONG *)sdebug    // developer info
};
```

FIG. 66

User Application

```
// load and run the event sequence
pDFN->DFNBeginSequenceNoMappingNoLog
  {snum, "d:\\HF.bin"};

//assign data to be passed
sndBuf = 25000;

// change the queue variable
pDFN->DFNChangeQueueVariable
{
  (char *)SymName,    // variable name
  (char *)sndBuf,     // new value
  (ULONG)sizeof sndBuf, // num bytes to write
  (ULONG *)sdebug    // developer info
};
```

FIG. 67

Ben Script

```
sub frame_type1
{
  $HFfrm = 'frame_type1';

  %qv = ('delay_qv' => [20000]);
  %qp = ();

  $image_cmd = {0x800000, 0x0};

  compile_init(@_, \%qp, \%qv, $HFfrm);

  Send($image_cmd);
  Delay('delay_qv');
  LoopKF(2, 0xAFF01);

  compile_finit();
}
```

FIG. 68

43

IMAGE DETECTION SUBSYSTEM

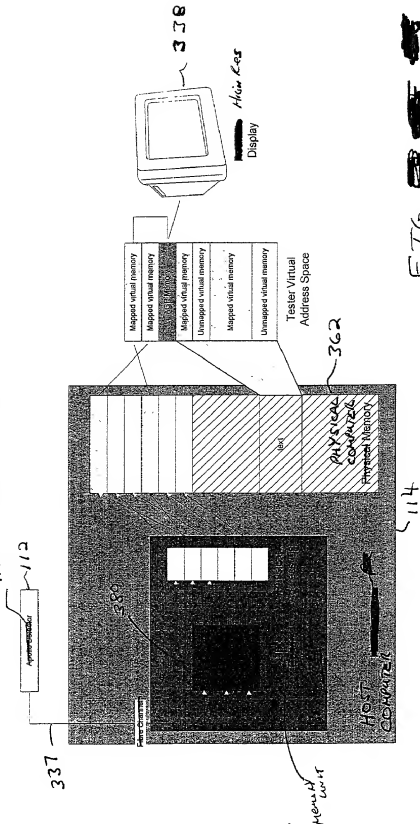


FIG.

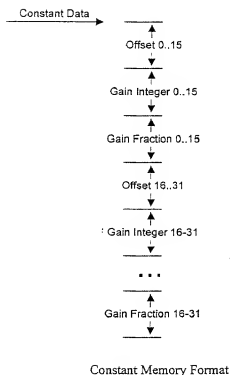
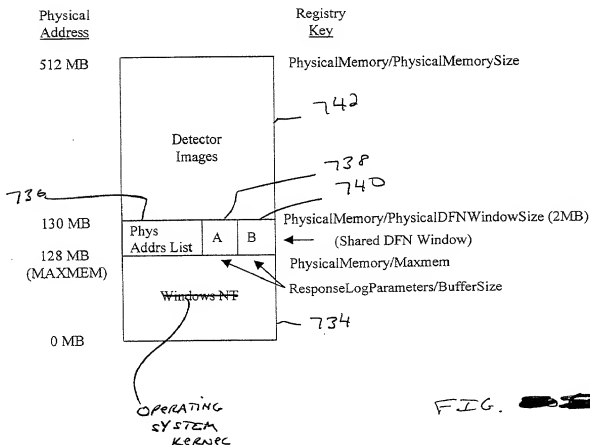
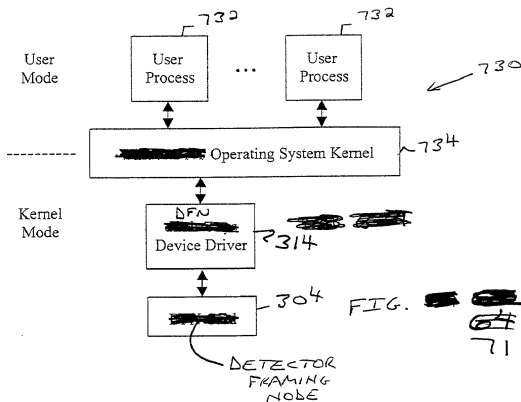


FIG. ~~70~~
70



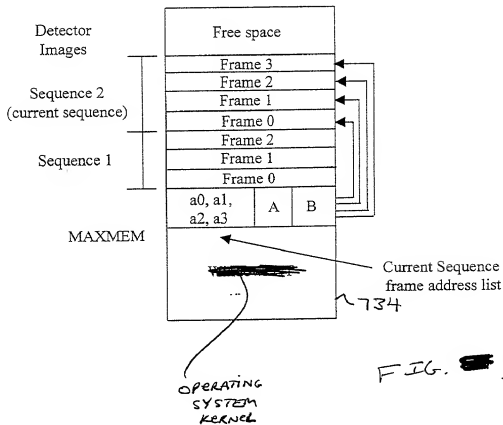


FIG. 73